FIG.1A

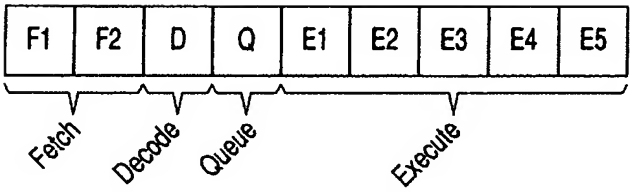

F1
F2
D
Q
E1
E2
E3
E4
E5
Fetch
Decode
Queue
Execute

FIG.1B

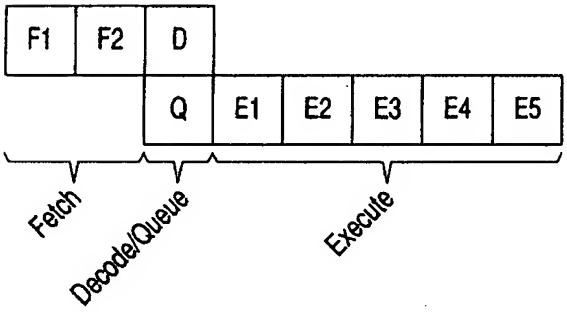
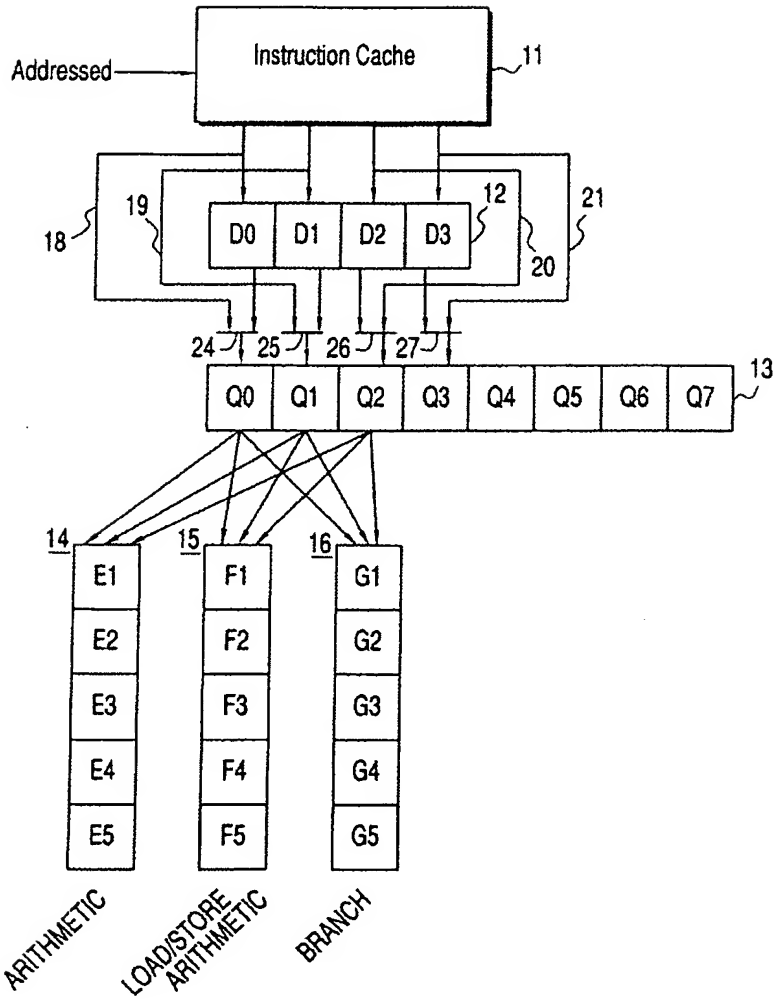

F1
F2
D
Q
E1
E2
E3
E4
E5
Fetch
Decode/Queue
Execute

FIG.2


Addressed
Instruction Cache
11
18
19
D0
D1
D2
D3
12
20
21
24
25
26
27
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
13
14
15
16
E1
E2
E3
E4
E5
F1
F2
F3
F4
F5
G1
G2
G3
G4
G5
ARITHMETIC
LOAD/STORE ARITHMETIC
BRANCH

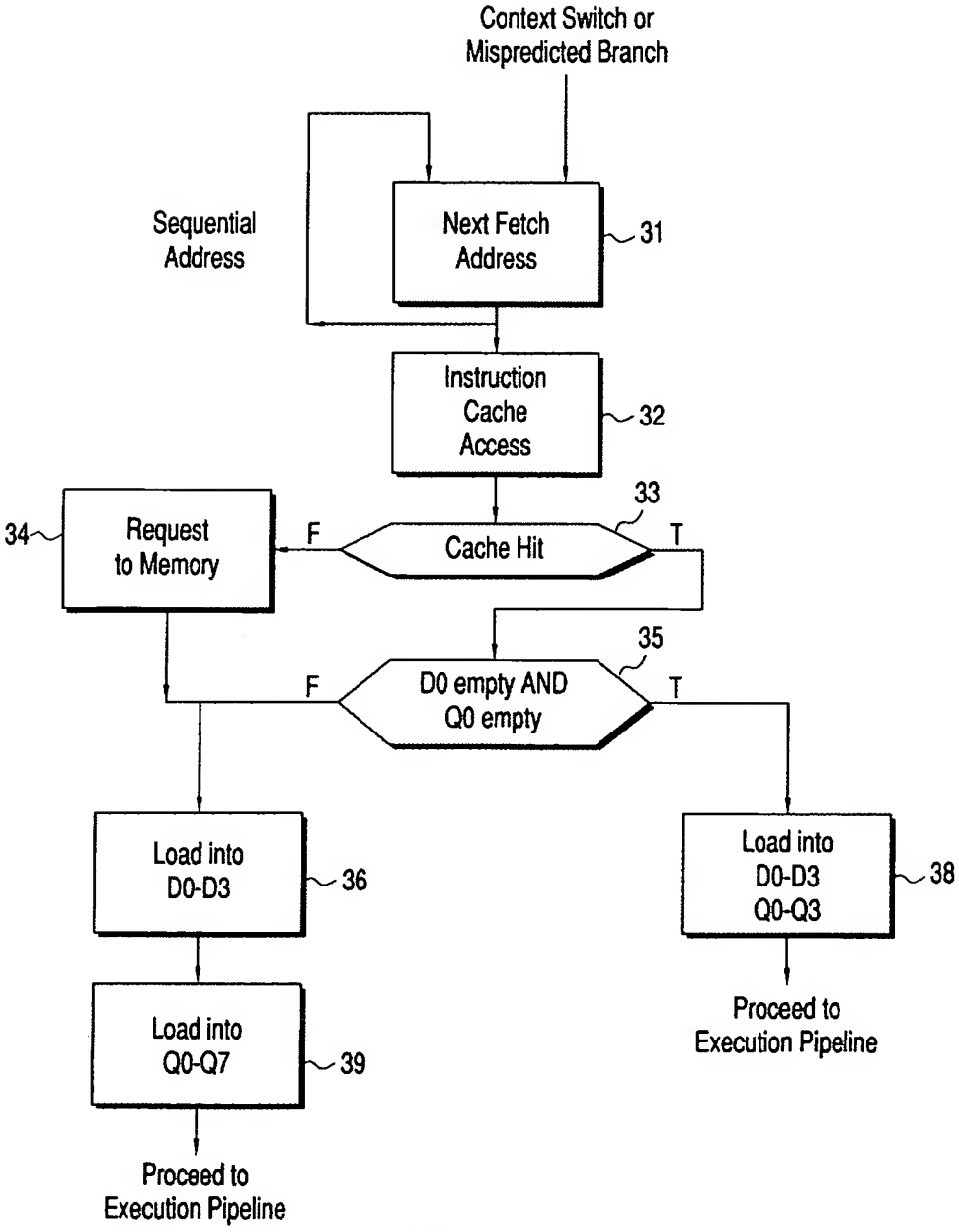

Context Switch or
Mispredicted Branch
Sequential
Address
Next Fetch
Address
31
Instruction
Cache
Access
32
33
34
Request
to Memory
F
Cache Hit
T
35
F
D0 empty AND
Q0 empty
T
Load into
D0-D3
36
Load into
D0-D3
Q0-Q3
38
Load into
Q0-Q7
39
Proceed to
Execution Pipeline
Proceed to
Execution Pipeline

FIG.3

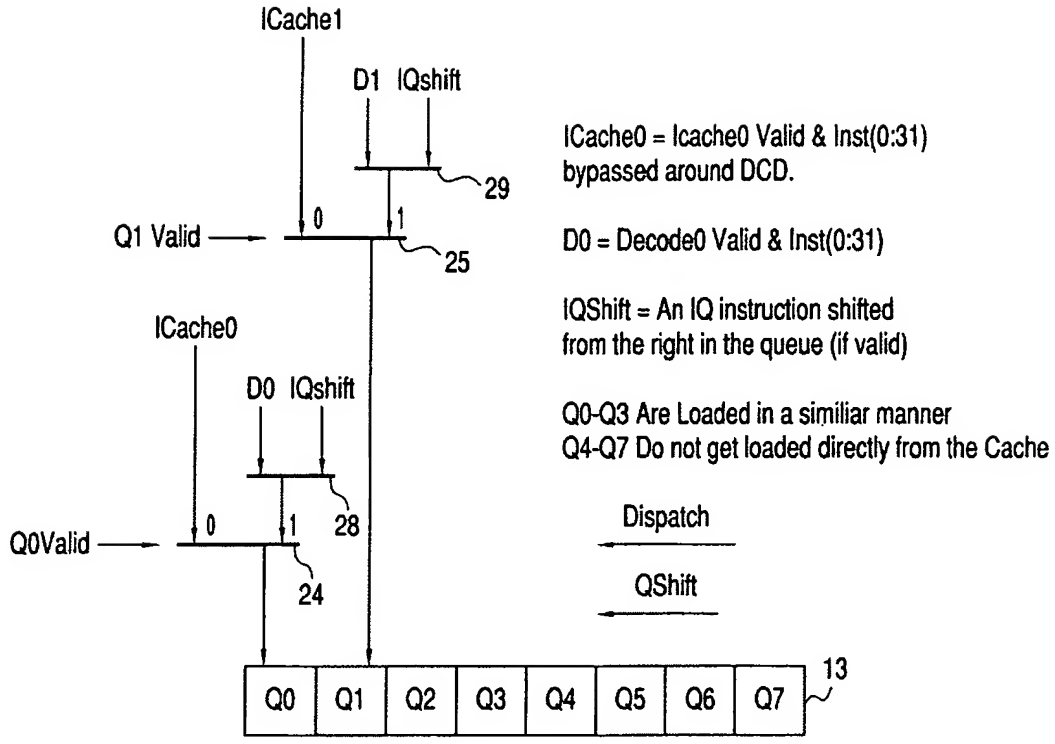

ICache1
D1 IQshift
29
Q1 Valid
0
1
25
ICache0
D0 IQshift
28
Q0Valid
0
1
24
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
13
ICache0 = Icache0 Valid & Inst(0:31) bypassed around DCD.
D0 = Decode0 Valid & Inst(0:31)
IQShift = An IQ instruction shifted from the right in the queue (if valid)
Q0-Q3 Are Loaded in a similiar manner
Q4-Q7 Do not get loaded directly from the Cache
Dispatch
QShift

FIG.4

Branch Predict Taken - Effect on IQ Instruction Flow

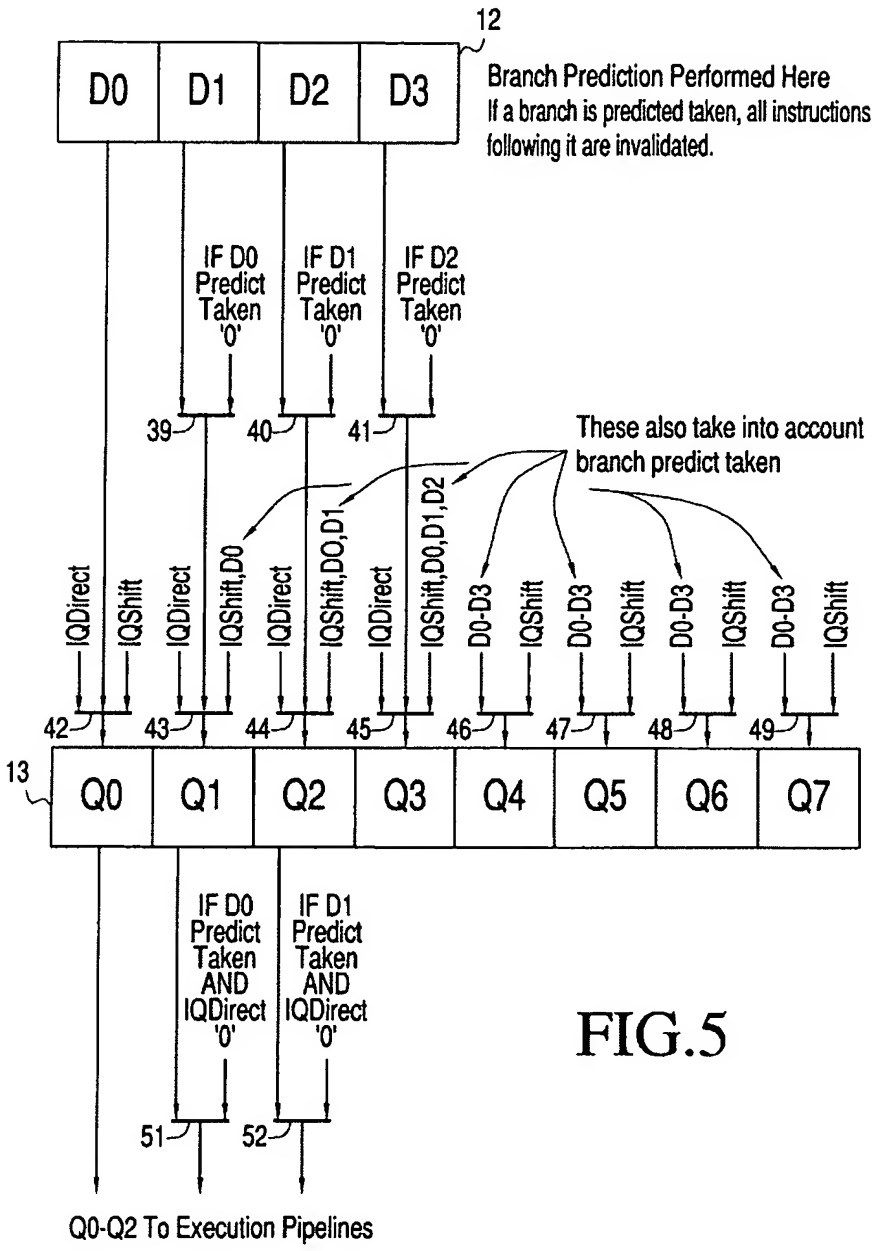


FIG.5